A Steel Surface Defect Detection Model Based on YOLOv7-tiny

Yayun Wang, Ye Tao, Wenhua Cui, and Lijia Shen

Abstract—**This article designs the PELAN structure based on the lightweight YOLOv7-tiny model for surface defect detection of hot-rolled steel strips. At the same time, the CA (Channel Attention) is embedded in the feature pyramid structure and the Slim-Neck structure is introduced at the neck. Through experimental result analysis, it is proved that the lightweight algorithm not only reduces model complexity, but also has better detection accuracy and processing speed. This article focuses on exploring the potential of lightweight YOLOv7-tiny models in steel surface defect detection. With the advancement of Industry 4.0, quickly and accurately detecting surface defects on steel strips has become crucial. However, traditional defect detection methods often face issues such as large computational requirements and poor real-time performance. Therefore, we have designed a novel PELAN structure that combines CA(Channel Attention)and Slim-Neck structure. The aim is to reduce model complexity while maintaining or even improving detection accuracy and processing speed.**

Index Terms— **Lightweight YOLOv7-tiny; PELAN structure; CA attention mechanism; Slim-Neck structure**

I. INTRODUCTION

THE earliest research in object detection was the algorithm for face detection proposed in the early 1990s [1]. Its development can be divided into two stages: the traditional object detection algorithm stage and the deep learning object detection algorithm stage. The traditional object detection algorithm mainly consists of three steps: region proposal. feature extraction. classification and regression. Although the traditional object detection algorithm has achieved good results in target recognition, it still has some shortcomings. such as limited recognition accuracy, high computational complexity, and slow processing speed. Until 2012, the rise of convolutional neural networks brought a new breakthrough in the field of object detection.

It is greatly improved the ability of network models to extract features of image targets, increasing accuracy while reducing detection time. Since then, deep learning-based object detection algorithms have become a hot topic in academia. According to different ways of target localization and classification, deep learning object detection algorithms are divided into two-stage and one-stage algorithms [2]. The two-stage algorithm divides the detection task into two separate stages. In the first stage, usually using region proposal networks to generate candidate regions. in the second stage, using classification networks to classify and localize these proposed regions.

The two-stage method usually performs better in terms of accuracy, but the design and training process is relatively cumbersome and the processing speed is slower. one-stage object detection algorithms have been proposed. To simplify the overall model design and tuning process and make the algorithm more suitable for real-time industrial and life scenarios. This kind of algorithm directly integrates the target localization and classification tasks into a single network, allowing simultaneous processing of target location and category information. Through regression technology, it directly generates the boundary box location information and probability distribution of target categories, greatly improving the inference speed.

In 2016, Redmon et al. [3] proposed the one-stage object detection algorithm YOLOv1. It adopted a completely new idea, abandoned the process of extracting candidate regions, and could quickly process a large number of images while maintaining high accuracy. However, it had weaker performance in dealing with small and dense targets. In 2016, Wei Liu et al. [4] proposed the SSD detection algorithm, which used multi-scale feature maps to detect targets of different sizes. It also introduced the concept of "default boxes" to adapt to various target detection tasks of different scales and densities. However, it also required large computational resources. In 2017, Redmon et al. [5] improved the YOLOv1 algorithm and proposed the YOLOv2 detection algorithm. By introducing Anchor Boxes for better adaptation to different shapes of targets, it improved the accuracy of predicting bounding boxes. However, when dealing with occlusion and dense targets, the accuracy of bounding boxes may decrease. In 2018, Redmon et al. [6] proposed the YOLOv3 object detection algorithm. It adopted the Feature Pyramid Network (FPN) to enable the network to obtain contextual information of different scales and improve multi-scale target detection capabilities. At the same time, it adopted a deeper Darknet-53 network structure to provide more powerful feature extraction capabilities, enhancing the detection effect of small targets. In 2020, Bochkovskiy et al. [7] proposed an improved version of YOLOv3 called YOLOv4. It adopted the CSPNet (Cross Stage Partial

Manuscript received Jan 2, 2024; revised Sep 18, 2024. This work was supported by the National Natural Science Foundation of China (62272093), and the Department of Education of Liaoning Province (LJKFZ20220197).

Yayun Wang is a lecturer at Anshan Normal University, Anshan, China. (e-mail: wangyayun1983@163.com).

Ye Tao is an associate professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (Corresponding author to provide phone: +86-133-0422-4928; e-mail: taibeijack@163.com).

Wenhua Cui is a Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: taibeijack@126.com).

Lijia Shen is a Postgraduate of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: maomaoyugougou898@163.com).

Network) module to enhance the performance of extracting network features. At the same time, it introduced the PANet (Path Aggregation Network) structure to fuse feature maps of different levels and improve the utilization efficiency of multi-scale features, greatly improving detection accuracy and speed. In 2020, Glenn Jocher et al. released the YOLOv5 series algorithm [10], which improved data augmentation strategies, increased diversity in training data, and improved model generalization capabilities. At the same time, they launched four versions to facilitate developers' selection and secondary development so that it can run on embedded devices and real-time applications. In 2022, Wang et al. [11] proposed the YOLOv7 series algorithm. They designed an efficient aggregation network architecture and introduced model reparameterization algorithms into the network architecture to effectively improve detection efficiency. This made it outperform previous YOLO series in terms of accuracy and speed. These models are generally large. It is necessary to build a more lightweight model that can be used on embedded devices. While increasing detection speed, it does not reduce detection accuracy.

II. Related Work

A. Lightweight YOLOv7-tiny Model

Model lightweighting aims to improve a model's inference speed and deployment efficiency in resource-constrained environments by reducing its complexity and number of parameters. Lightweight models can perform real-time object detection and recognition on mobile devices, embedded systems, and other resource-limited environments. They can also be more easily deployed to different devices and systems, adapting to various application scenarios and requirements, and enhancing the model's flexibility and versatility.

B. YOLOv7-tiny Model

The YOLOv7-tiny model is a version of the YOLOv7 series designed specifically for devices with limited computational resources. It is an efficient lightweight object detection algorithm with a structure shown in Fig.1. It adopts a more compact network architecture and optimized training strategies to reduce the model's parameter count and computational load, enabling real-time operation on embedded devices and mobile terminals.

The YOLOv7-tiny model also consists of four parts. Input, Backbone, Neck, and Head. In the Backbone section, a relatively simple ELAN-E structure is used instead of the ELAN structure in YOLOv7. During sampling process, only max pooling is used for feature extraction, and the original convolution operation is eliminated. An optimized feature pyramid SPPC structure is designed to obtain more feature map information. In the Neck section, the PANet structure is still used to aggregate features. In the Head section, standard convolution is used instead of reparameterization modules to adjust the number of channels. Although the YOLOv7-tiny model has advantages in terms of lightweightness and detection speed compared to the YOLOv7 model, it still has the disadvantages of complex structure in the Backbone and Neck sections. Therefore, this chapter uses a more lightweight solution to improve the YOLOv7-tiny model. Under the premise of ensuring detection accuracy and speed, the model's computational complexity and parameter count are further reduced.

C. PELAN Structure

The ELAN-E structure in the YOLOv7-tiny model consists of multiple traditional convolutions densely connected, leading to an increase in computational complexity. This article redesigned the ELAN-E structure and proposed the PBH module. The PBH module consists of *PConv* (Partial Convolution), BN, and the Hardswish activation function. Compared to traditional 3×3 convolutions, *PConv* achieves better feature representation

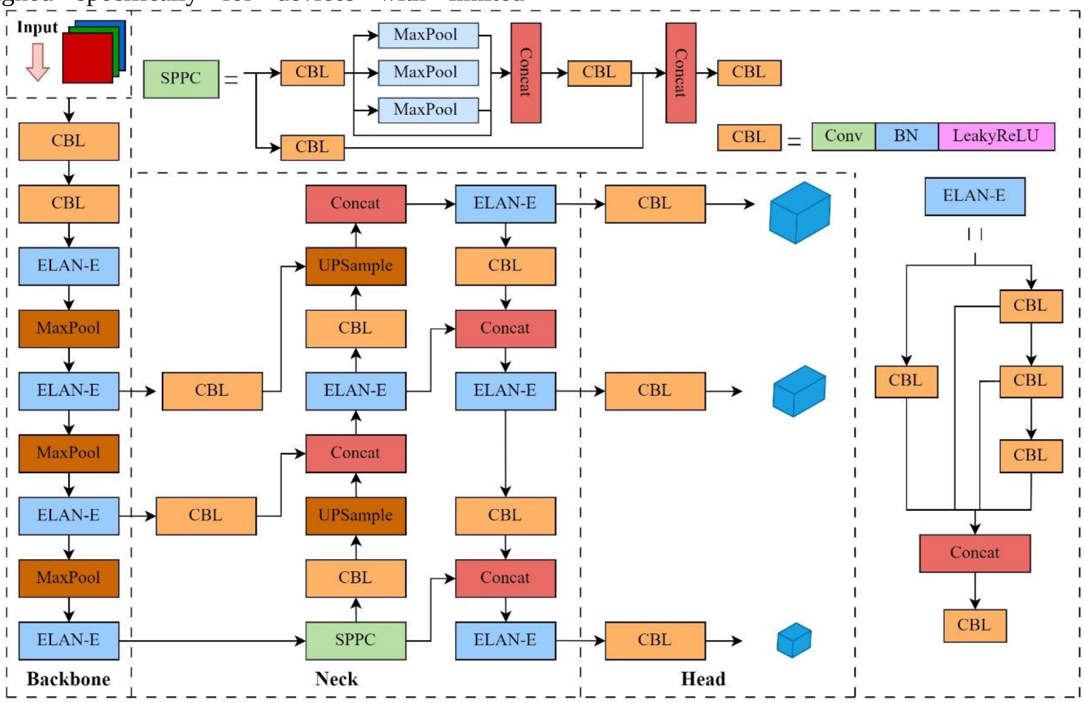

Fig. 1 Structure diagram of YOLOv7-tiny

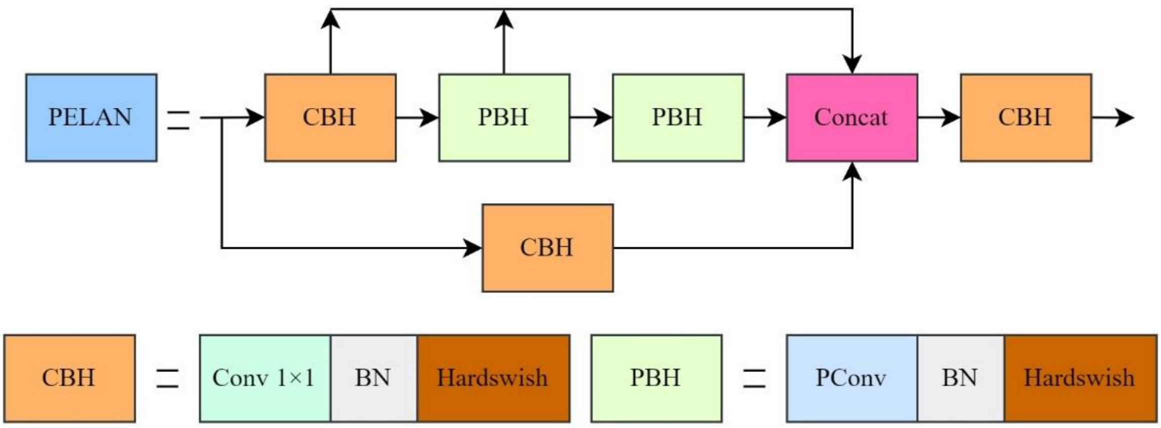

Fig.2 Structure diagram of PELAN

learning without introducing additional parameters and computational complexity. At the same time, using the Hardswish activation function greatly reduces the number of memory accesses, improves detection speed, and reduces computational costs. The PELAN structure is composed of CBH and PBH modules, replacing the original CBL module in ELAN-E. The PELAN structure is shown in Fig. 2. The design process is as follows.

1) PConv Partial Convolution

PConv [12] is a novel convolution operation mode. It can effectively extract spatial features, reduce redundant calculations and memory accesses, and solve the problem of traditional convolution on high frequency access delay. *PConv* partial convolution not only has the advantage of high computational efficiency, but also has relatively few model parameters. *PConv* partial convolution can express the model effectively with fewer parameters, thereby reducing the computational and memory usage of the model. In addition, it helps to improve the performance of small target detection in the model. Its structure is shown in Fig.3.

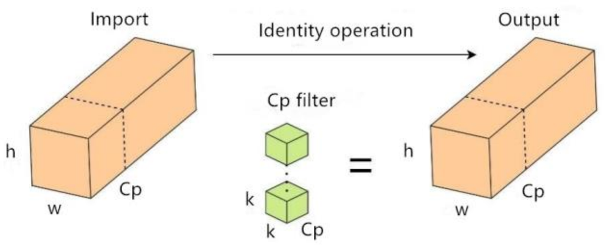

Fig.3 Structure diagram of Pconv

In the *PConv* Partial Convolution, it extracts spatial features by applying regular convolution to a subset of input channels while keeping the size of other channels unchanged. It avoiding redundancy in the feature map generated by different channels. When dealing with memory-continuous or regular accesses, it treats the first or last continuous channel as the representative of the entire feature map for computation. At this time, the computational complexity of *PConv* is calculated as shown in Equation (1).

$$h \times w \times k^2 \times C_p^2 \qquad (1)$$

For the typical partial ratio r = 1/4, the computational complexity of *PConv* is only 1/16 times that of regular Conv. Additionally, *PConv* has less memory accesses and its computational complexity is shown as follows:

$$h \times w \times 2C_p + k^2 \times C_p^2 \approx h \times w \times 2C_p \qquad (2)$$

In this formula, h represents the height of the channel, w represents the width of the channel. Cp refers to the number of continuous network channels. K stands for the filter size. Moreover, *PConv* can better utilize the computing capabilities of devices and demonstrate good performance in spatial feature extraction. Therefore, using *PConv* can enhance the generation capability of feature maps while maintaining network lightweight. *PConv* can better providing advantages for model training and inference.

2) Hardswish Activation Function

Due to limited expressive power of linear models. Neural networks incorporate nonlinearity by using activation functions. Enriching the expressive capacity of deep neural networks and enhancing model accuracy and performance. In the CBL module of ELAN-E, the LeakyReLU activation function is employed, a variant of the ReLU function. LeakyReLU allows a small portion of negative inputs to pass, with a slope in the negative region, adjustable to fit different problems and data. This prevents "dead neurons" from never being activated during training. The LeakyReLU function is expressed in formula (3).

$$Leaky\,\mathrm{Re}\,LU(x) = \begin{cases} x, (x \geq 0) \\ \alpha x, (x < 0) \end{cases} \qquad (3)$$

The disadvantage of the LeakyReLU activation function is that it requires setting an additional parameter α. It increases the complexity of the model. Additionally, when α is improperly set, it can lead to excessive or insufficient neuron outputs. It impacting the model's expression capabilities. The Hardswish activation function [13] adopted in this article is a smooth non-monotonic activation function. This function can improve the accuracy of neural networks while utilizing the segmented function. It can reduce memory access times and computational costs, making calculations faster and more conducive to training. This allows the network to learn more expressive features. The Hardswish function is expressed as Equation (4).

$$Hardswish(x) = \begin{cases} 0 & x \leq -3 \\ x & x \geq 3 \\ x \cdot (x+3)/6 & -3 < x < 3 \end{cases} \qquad (4)$$

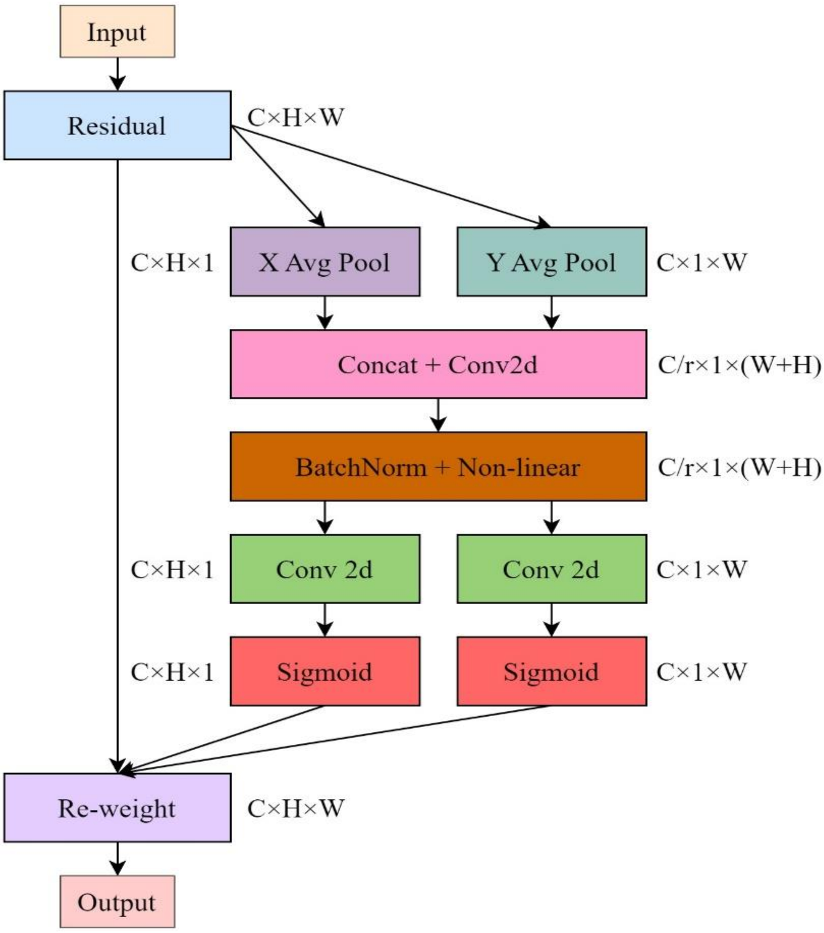

Fig.4 Structure diagram of CA

D. *CA Attention Mechanism*

To enhance YOLOv7-tiny's focus on surface defect location in hot-rolled steel strips and reduce background noise. We integrate the CA attention mechanism [14] into the feature pyramid of the model. This spatial-based attention mechanism incorporates spatial information as an attention factor in the inference layer of the feature extraction network. This allows the model to prioritize regions with high feature weights, making it more sensitive to defect location information. The structure of the CA attention mechanism is illustrated in Fig.4.

The CA attention mechanism first performs two global average poolings on the input feature map of size C×H×W, generating feature maps of size C×H×1 and C×1×W. It then transposes the width and height dimensions of these two feature maps and fuses them using a stacking operation. Afterwards, a 1×1 convolution operation is performed on the fused feature map. The feature is further processed using a BN layer and a nonlinear activation function to obtain a new feature map of size C/r×1×(W+H). Next, along the spatial dimension, the new feature map is split into two feature maps for width and height, and transposed to restore the original width and height dimensions, generating two feature maps of sizes C×H×1 and C×1×W. Then, 1×1 convolutions are used to adjust the number of channels to adapt to attention calculations. Finally, the Sigmoid activation function is used to obtain attention weights on the width and height dimensions. These attention weights are multiplied and added to the original feature map to generate a feature map with coordinate awareness.

The CA attention mechanism makes full use of the characteristics of surface defects on hot rolled steel strips. The connection between space and channel effectively extracts key information about the location of defects. This improves the network's positioning accuracy and recognition effectiveness for defect information.

E. *Slim-Neck Structure*

In YOLOv7-tiny model, the Neck is responsible for connecting the Main Body and the Head to fuse and process the features. This can improve the detection accuracy and efficiency. To further reduce model complexity and parameter count, this chapter introduces a Slim-Neck structure to optimize the Neck section of the YOLOv7-tiny model. This structure consists of GSConv convolution and VoVGSCSP cross-level networks. While maintaining accuracy, it can reduce model complexity.

1) GSConv Ghost Convolution

In traditional convolutional neural networks, spatial information in feature maps needs to be converted into channel information. However, each spatial compression and channel expansion operation can cause partial loss of semantic information. The GSConv Ghost Convolution [15] enables information exchange and recombination between channels without losing representational ability. It can better balance model accuracy and speed, with its time complexity calculated as shown in Equation (5). Its structure is illustrated in Fig. 5.

$$T_{GSConv} = O\left(\frac{W \times H \times K_1 \times K_2 \times C_2}{2} \times (C_1 + 1)\right) \quad (5)$$

In the equation, W and H represent the width and height of the output feature map respectively. K1×K2 denotes the size of the convolution kernel. C1 represents the number of input feature map channels. C2 represents the number of output feature map channels.

GSConv starts with a conventional convolution on C1-channel input feature maps, reducing the number of channels to C2/2. Then, it applies depth wise separable convolution DWConv[16] independently to each channel, maintaining the same number of channels. The feature maps after conventional and depthwise separable convolutions are concatenated, followed by a shuffle operation to rearrange the feature channels. The final output has C2 channels, enhancing semantic information and improving feature representation ability.

2)VoVGSCSP Cross-level Part Network

The cross-level part network of VoVGSCSP adopts the method of one-time aggregation to construct it. It can effectively fuse the information between feature maps of different stages. The GS bottleneck layer structure is further enhanced by stacking GSConv Ghosting Convolution to improve the network's ability to process features. Enhance the nonlinear expression of features and improve the reuse of information. By introducing the cross-level part network of VoVGSCSP, it can reduce the computational complexity and the complexity of network structure. It improve the utilization efficiency of features and network performance. It maintain the detection accuracy. Its structure is shown in Fig. 6.

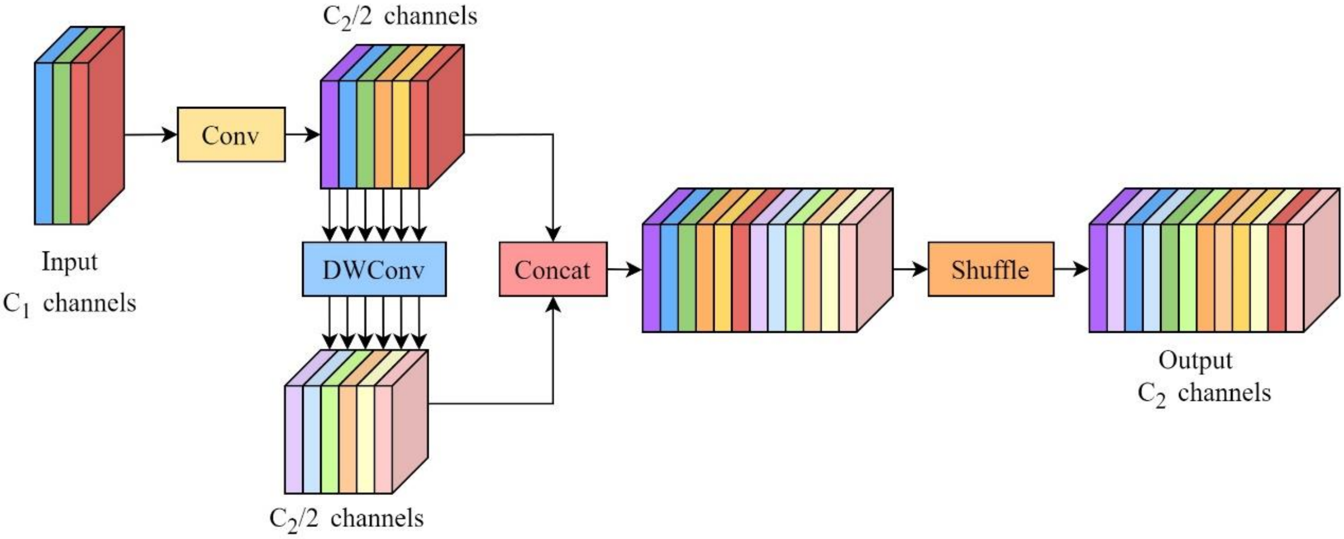

Fig.5 Structure diagram of GSConv

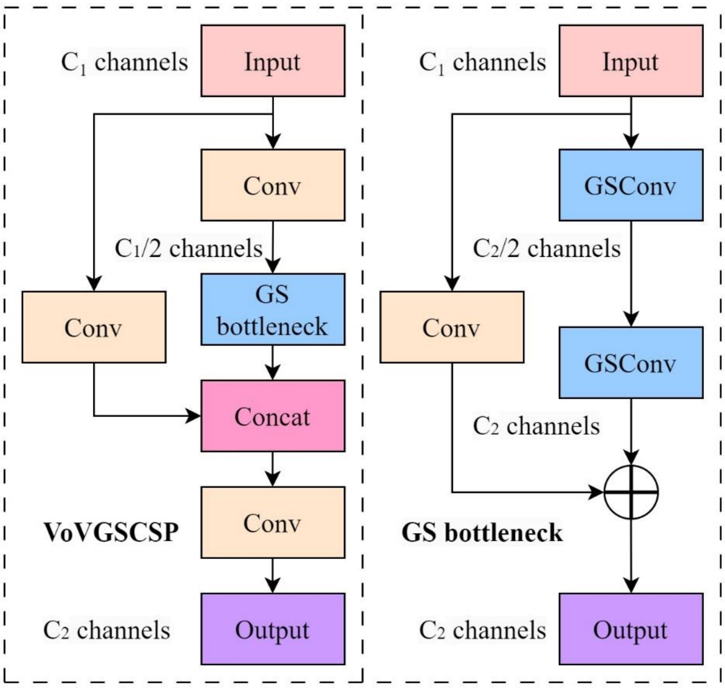

Fig.6 Structure diagram of VoVGSCSP

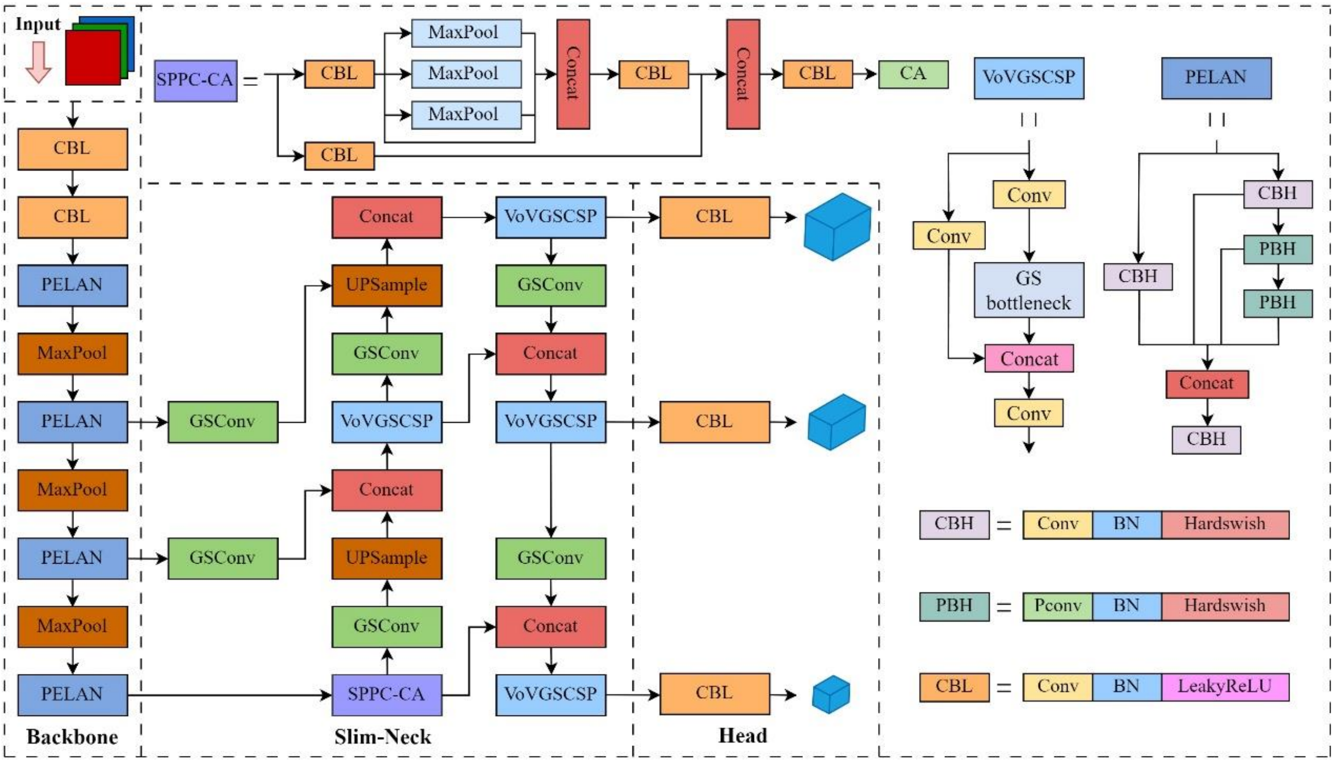

Fig.7 Structure diagram of Lightweight YOLOv7-tiny

III. LIGHTWEIGHT YOLOV7-TINY MODEL EXPERIMENT

A. Lightweight YOLOv7-tiny Model

The lightweight YOLOv7-tiny model structure is shown in Fig.7. This article first proposes the PELAN structure, which replaces the original model's ELAL-E structure. While maintaining lightweight, it enhances the feature map generation capability. To address the issue of edge position information loss of surface defects on hot-rolled steel strips in complex scenes. We incorporate a CA attention mechanism into the feature pyramid to enhance the model's focus on the defect edge regions and improve detection accuracy. Finally, the Slim-Neck structure is introduced at the neck, reducing the model's computational and parameter counts while maintaining detection accuracy.

B. Evaluation Indicators

The experimental hardware environment parameters used in this article are as follows in TABLE I.

The software environment for this experiment uses the PyTorch 1.12.1 framework based on Python 3.9 to run the code. The cosine annealing learning algorithm is selected to update the learning rate. The learning rate is set to 0.01. The training batch size is 8. The total training epoch is 300. The stochastic gradient descent optimizer SGD is used to iterate the network parameters. The weight decay coefficient is 0.0005. The momentum factor is 0.937. In order to evaluate the performance of the lightweight YOLOv7-tiny model in this article, four evaluation metrics. including mean average precision (mAP), model parameter quantity (Params), model computation quantity (GFLOPs), and frame per second (FPS) to measure the real-time detection speed of the model.

TABLE I
Experimental environment

Experimental environment	Version model
Operating System	Windows10
CPU	Intel i7-10700F CPU @ 2.90GHz
GPU	NVIDIA GeForce RTX 3070
video memory	8GB
Deep Learning Framework	PyTorch 1.12.1
Development Language	Python 3.9
Development tools	PyCharm 2021.3.2

IV. Experiments and Result Analysis

A. Comparative Analysis of Detection Effects of Lightweight Models on Surface Defects of Hot-Rolled Steel Strips Using Confusion Matrices

Through confusion matrices, a comparative analysis can be conducted on the detection effects of lightweight models on surface defects of hot-rolled steel strips. Fig.8 shows the confusion matrices generated for six categories after training. It can be seen from the diagonal of the matrix that most defect categories can be correctly predicted, indicating that the model has a good detection performance.

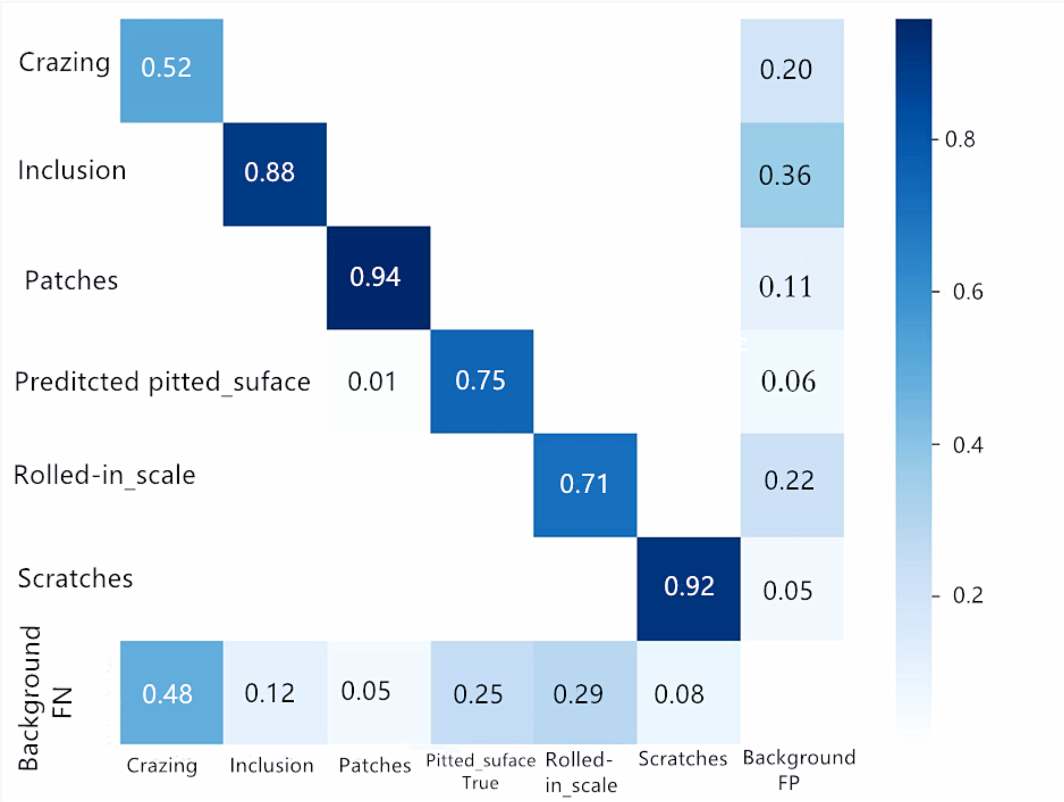

Fig.8 Confusion Matrix of defect detection

TABLE II
Ablation experiment

NO.	PELAN	SPPC-CA	Slim-Neck	Map,%	Params,M	FLOPs,G	FPS
N1	-	-	-	71.1	6.2	14.2	93
N2	✓	-	-	72.6	5.1	16.7	108
N3	-	✓	-	71.8	6.2	14.3	90
N4	-	-	✓	73.3	5.5	11.9	116
N5	✓	-	✓	74.7	4.7	15.2	134
N6	✓	✓	✓	75.2	4.7	15.4	127

TABLE III
Comparative experiment

Method	Map, %	Params, M	FLOPs, G	FPS
YOLOv5s	69.6	7.1	16.4	82
YOLOv6s	70.3	18.5	45.1	86
YOLOv7-tiny	71.1	6.2	14.2	90
DCN-YOLO	76.9	16.3	39.5	93
Ours	75.2	4.7	15.4	127

B. ablation experiments

To verify the effectiveness of the lightweight improvements proposed in this chapter, a series of ablation experiments were conducted. The experiments used YOLOv7-tiny as the baseline model, and the results are presented in TABLE II. A total of N1-N6 six sets of ablation experiments were conducted to verify the effectiveness of each improvement point. "✓" represents the adoption of the improvement point. "-" represents the absence of the improvement point.

In the table, N1 group uses the original YOLOv7-tiny model for experiments. The experimental results show that the mAP value of the baseline model YOLOv7-tiny reaches 71.1%. The model parameters are 6.2M. The computational complexity is 14.2G. The FPS value is 93. There is still room for improvement in terms of lightweight and detection speed. N2 group experiments replace the original ELAN-E structure in the YOLOv7-tiny backbone network with the PELAN structure designed in this chapter. At this time, the mAP value reaches 72.6%. The model parameters are 5.1M. The computational complexity is 16.7G. Compared with the original YOLOv7-tiny model, although the computational complexity increases but the mAP value improves by 1.5%. The model parameters decrease by 0.9M. The FPS value increases to 108. This proves that the PELAN structure is effective for model lightweighting. N3 group experiments embed the CA attention mechanism into the SPPC structure. While slightly increasing computational complexity, the mAP value improves to 71.8%. Compared with the original model, the mAP value improves by 0.7%. The algorithm's detection performance for small targets on the surface defects of hot-rolled steel strips is improved. N4 group experiments adopt the Slim-Neck structure. It reduces model parameters and computational complexity while increasing the mAP value by 2.2% and FPS value to 116 compared with the original model. N5 group experiments combine the improved points of N2 and N4. The mAP value increases to 74.7%. The FPS value increases to 134. Demonstrating that these two structures can significantly improve detection accuracy and speed while lightweighting the model. N6 is a combination of all three improved points. N6 with a mAP value of 75.2%. Model parameters of 4.7M. Computational complexity is 15.4G. Compared with the original YOLOv7-tiny model, although the computational complexity of the model increases by 1.2G. The mAP value improves by 4.1%. The model parameters decrease by 1.5M. The FPS value increases to 127. This indicates that the algorithm constructed in this chapter can further lightweight itself while effectively improving detection accuracy and speed for small targets.

To further validate the effectiveness of the lightweight YOLOv7-tiny model proposed in this article. The algorithm in this article is compared with the YOLOv5s, YOLOv6s, YOLOv7-tiny, and DCN-YOLO target detection algorithms. The results are summarized in TABLE III. Through experimental results, it can be seen that the lightweight YOLOv7-tiny model has smaller computational complexity and parameter quantity. Its mAP value is 5.6% higher than that of the YOLOv5s model and its FPS is 45 higher. Compared with the YOLOv6s model, the mAP value is increased by 4.9%. The FPS is increased by 41. Compared with the original YOLOv7-tiny model, the mAP value is increased by 4.1%. The FPS is increased by 37. Compared with DCN-YOLO [52], although the accuracy is slightly lower, the computational complexity and parameter quantity are significantly reduced. The FPS has been increased to 127. It can be concluded that the lightweight model can achieve good detection accuracy and speed while reducing computational complexity.

V. CONCLUSION

In this article, we introduce a lightweight YOLOv7-tiny model. The PELAN structure is first proposed, which consists of PBH and CBH modules built using PConv partial convolution, BN layer, and Hardswish activation function. This structure offers better feature representation learning while reducing model complexity and computational cost compared to the original ELAN-E structure. Next, we embed the CA attention mechanism into the feature pyramid to enhance focus on edge position regions and improve defect recognition. Finally, we optimize the neck network of the YOLOv7-tiny model using the Slim-Neck structure. It improving feature utilization and detection efficiency while further reducing computational complexity and parameter count. After ablation and comparison experiments, the lightweight YOLOv7-tiny model achieves a mAP value of 75.2%, with a 1.5M reduction in parameters compared to the original model. Although computational complexity is slightly increased, the mAP value is improved by 4.1%, and the FPS value is boosted to 127. This lightweight model offers detection accuracy and FPS improvements while reducing model complexity, providing valuable reference for deployment on mobile devices and edge devices.

REFERENCES

[1] M. A. Turk and A. P. Pentland. "Recognition in Face Space," Intelligent Robots and Computer Vision IX: Algorithms and Techniques, pp43-54, 1991.

[2] Yuan Jiaojiao, Hu Yongli, Sun Yanfeng, et al. A Survey of Small Target Detection Methods Based on Deep Learning [J]. Journal of Beijing University of Technology, vol.47, no.3, pp293-302, 2021.

[3] J. Redmon, S. Divvala, R. Girshick. "You only look once: Unified, real-time object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp779-788, 2016.

[4] W. Liu, D. Anguelov, D. Erhan, et al. "Single shot multibox detector," Computer Vision–ECCV 2016:14th European Conference, Amsterdam, pp21-37, 2016.

[5] J. Redmon and A. Farhadi. "YOLO9000: better, faster, stronger," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp7263-7271, 2017.

[6] J. Redmon and A. Farhadi. "Yolov3: An incremental improvement", arXiv preprint arXiv:1804.02767, 2018.

[7] A. Bochkovskiy, C. Y. Wang and H. Y. M. Liao. "Yolov4: Optimal speed and accuracy of object detection", arXiv preprint arXiv:2004. 10934, 2020.

[8] C. Wang, H. M. Liao, Y. Wu, et al. "CSPNet: A new backbone that can enhance learning capability of CNN", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp390-391, 2020.

[9] J. Yang, X. Fu, Y. Hu, et al. "PanNet: A deep network architecture for pan-sharpening", Proceedings of the IEEE Conference on Computer Vision, pp5449-5457, 2017.

[10] G. Yang, W. Feng, J. Jin, et al. "Face mask recognition system with YOLOV5 based on image recognition", Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), pp1398-1404, 2020.

[11] C. Wang, A. Bochkovskiy and H. M. Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object

detectors," Proceedings of the IEEE/CVF Conference on Computer Vision and *Pattern Recognition*, pp7464-7475, 2023.
[12] J. Chen, S. Kao, H. He, et al. "Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp12021-12031, 2023.
[13] A. Howard, M. Sandler, G. Chu, et al. "Searching for mobilenetv3", *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp1314-1324, 2019.
[14] Q. Hou, D. Zhou and J. Feng. "Coordinate attention for efficient mobile network design", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp13713-13722, 2021.
[15] H. Li, J. Li, H. Wei, et al. "Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles". arXiv, 2022: *https://arxiv.org/abs/2206.02424*.
[16] F. Chollet. "Xception: Deep learning with depthwise separable convolutions", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp1251-1258, 2017.